Amendments to the Claims:

1. (Currently amended) A system for developing distributed applications over a network of computing units, the system comprising:

    a.   a plurality of component programs installed over the network of computing units to create the distributed application, <u>wherein said component program receives data presented at its input ports, processes said data and writes the results to its output ports</u>;

    b.   a plurality of data stores on one or more of the computing units <u>in which the distributed application is composed externally by specifying the component programs that make up the application, the communication routes between the component programs, and the nodes on which the component programs are to run;</u> and ~~comprising a plurality of routes for data transfer between the component programs and a plurality of parameters for configuring the component programs; and~~

    c.   ~~a plurality of controller programs running on one or more of the computing units in the network for interacting with the component programs and for interacting with other controller programs to send and receive data by referring to routing information from the data stores.~~ <u>multiple controller programs running on multiple computing units in the network for transmitting data to said input ports of the component program and receiving data from said output ports of the component programs and for transmitting and receiving data with controller programs and disseminating said routing information to other controller programs</u>.

Claim 2-5.   (Canceled)

2

Claim 6. (Original) The system as recited in claim 1 wherein the component programs are adaptors for communicating with external applications that are not installed within the system.

Claim 7-8. (Canceled)

Claim 9. (Original) The system as recited in claim 1, wherein the data store can be replicated for high availability on a multiplicity of computing units.

Claim 10-25. (Canceled)

Claim 26. (Currently amended) A method for developing distributed applications over a network of computing units, with ~~one or more computing units having a controller program running on it~~ multiple controller programs running on multiple controller units, the method comprising steps of:

    a. customizing component programs;

    b. registering the component programs;

    c. composing ~~the~~ a distributed application externally in a data store, wherein said distributed application is composed by specifying the component programs that make up the application, the communication routes between these component programs, and the nodes on which the component programs are run;

    d. running multiple controller programs on multiple computing units in the network for transmitting data to input ports of the component program and receiving data from output ports of the component programs and for transmitting and receiving data with controller programs and disseminating said routing information to other controller programs.

    e. receiving said transmitted data at the input port of a component program, processing said data within said component program and writing the results to the output ports of the component program; and

f.   checking the connectivity and resources.

g.   executing the distributed application.

Claim 27.   (Canceled)

Claim 28.   (Currently amended) The method as claimed in claim 26, wherein the registering comprises the steps of:

a.   installing component programs on the computing units;

b.   specifying the external resources required by the component programs;

c.   specifying the input and output channels of the component programs; and

d.   making the component programs accessible to ~~one or more the participating computing units in the network~~ said computing units.

Claim 29.   (Currently amended) The method as claimed in claim 26, wherein ~~the~~ composing the distributed application comprises the steps of:

a.   choosing a subset of component programs from a set of component programs;

b.   adding and specifying the routes between the component programs;

c.   specifying the computing unit on which the component program is to be run;

d.   defining run time attributes of the component programs;

e.   defining various attributes of the routes; and

f.   storing the composed distributed application in a data store.

Claim 30.   (Canceled).

Claim 31.   (Original) The method as claimed in claim 29 wherein the defining attributes of the routes comprises defining the route type as peer-to-peer.

Claim 32.   (Currently Amended) The method as claimed in claim 26, wherein the composition of the distributed application is done using a <u>graphical user interface</u> (GUI).

Claim 33.   (Original) The method as claimed in claim 26, wherein the checking the connectivity and resources comprises steps of:

a.   checking if all controller programs are already running on the computing units in the network;

b.   checking if all the component programs are installed on the computing units on which they are specified to launch; and

c.   installing the component programs on the computing units in case they are not already installed.

Claim 34.   (Currently amended) The method as claimed in claim 26, wherein the executing <u>the distributed application,</u> comprises the steps of:

a.   receiving of data for the component program by controller programs;

b.   collecting of data by the appropriate component program from the controller program;

c.   processing of data by the component program;

d.   receiving of processed data from the component program by the controller program; and

e.   <s>passing of</s> <u>transmitting</u> the data to the next component program or a plurality of next controller programs based on the routing information stored in the routing table by the controller program interacting with it.

Claims 35-37. (Canceled)

Claim 38.   (Original) The method as recited in claim 34, wherein the passing of the processed data to the next component program based on the routing information stored in the routing table by the controller program comprises steps of:

a.   tagging the processed data with the name of the destination component program;

b.   placing the processed data by the controller program in the message bus for the controller program on whose node the destination component is installed for fetching, where the routing table specifies a hub/spoke route type; and

c.   sending the processed data by the controller program directly to the controller program on whose computing unit the destination component program is installed, where the routing table specifies a peer to peer route type.

Claims 39 - 48. (Canceled)